

jMax Phoenix

Le jMax Nouveau est Arrivé

Maurizio De Cecco <maurizio@jmax-phoenix.org>

François Dechelle <francois@jmax-phoenix.org>

Enzo Maggi <enzo@jmax-phoenix.org>

The MAX History



- Patcher (198x)
- Max Opcode (198x)
- Max ISPW (1989)
- Max/FTS (1994)
- Pure Data (1996)
- Max/MSP (1997)
- jMax (1996)
 - jMax GPL (1999)
 - RIP 2002

Why jMax Phoenix ? (1/2)

- Ever wanted to go back in time to fix something that went horribly wrong ?

cvsv -root \$(JMAXCVS) checkout -D 1 september 1999

Why jMax Phoenix (2/2)

- Because it is there and because we are there
 - .. and we can still read and maintain the code after ten years
- Because it runs well on modern multi core PC
- Because it uses an high end, free, portable and efficient Java graphic environment
- Because its architecture has a lot of headroom for evolution
- Because choice is a good thing

What is jMax Phoenix

- A community project
 - Not controlled by a single institution
- Started from some of the original jMax developers
- Project priorities
 - A modern UI
 - Integration into the modern Linux audio environment
 - Scalability !!!

The User Interface (1/5)

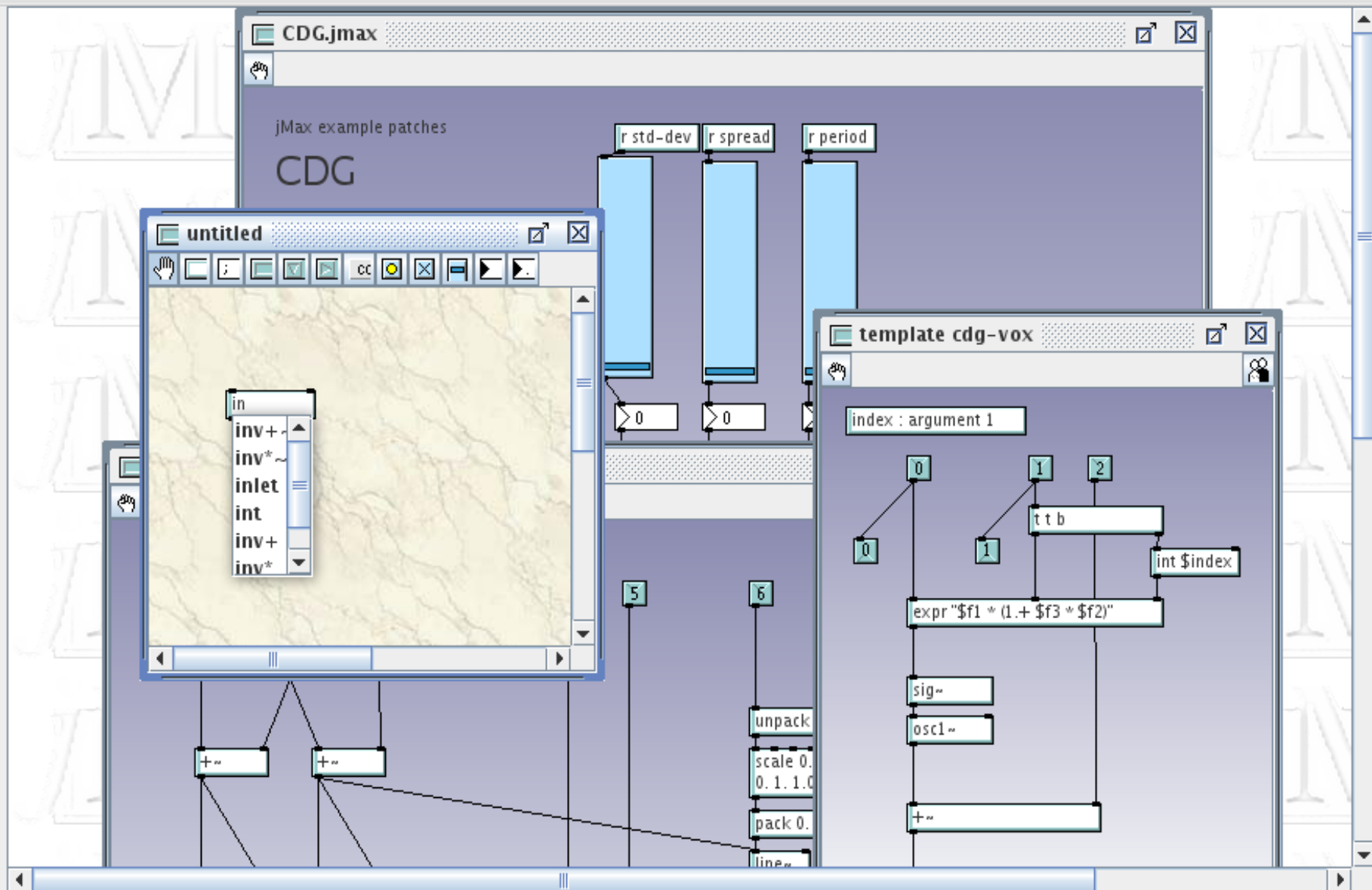
- A flexible container based framework
 - Separation of container from tools and editors
 - Multi-container architecture
- Classic container
 - Culturally backward compatible
 - Multi windows
- Multi-splitted container
 - User customizable
 - Allows IDE style interfaces

The User Interface (2/5)

- New User interface element
 - Patch browser
 - Configuration inspector (debug tool)
 - Global status bar
 - Auto completion for objects
- Basic cleanups
 - Object selection, editing, adding
 - Standard behaviour (mouse and key bindings)
 - Contextual menu, window menus
 - Automatic default configuration

jMax

- CDG.jmax
 - cdg-vox 0
 - cdg-vox 1
 - cdg-vox 2
 - cdg-vox 3
 - rev42 CDG
 - feed_in
 - feed_bac
 - out
 - front \$arg
- untitled

**jMax**

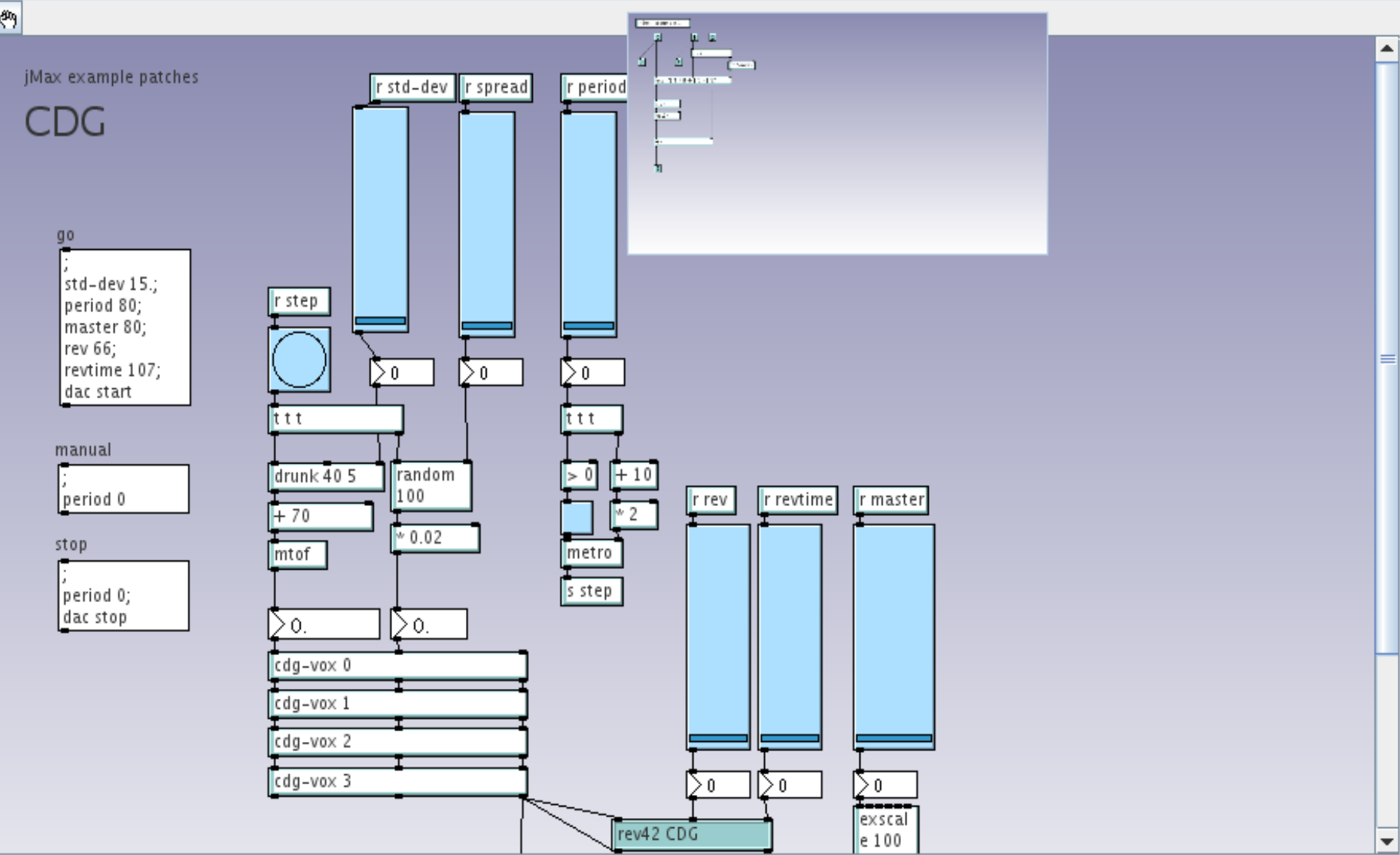
jMax Console ✕

Copyright (C) 2008 Maurizio De Cecco, Enzo Maggi, Francois Dechelle
Copyright (C) 1994, 1995, 1998, 1999 IRCAM - Centre Georges Pompidou
This is free software with ABSOLUTELY NO WARRANTY.
See file LICENSE for more details.

jMax

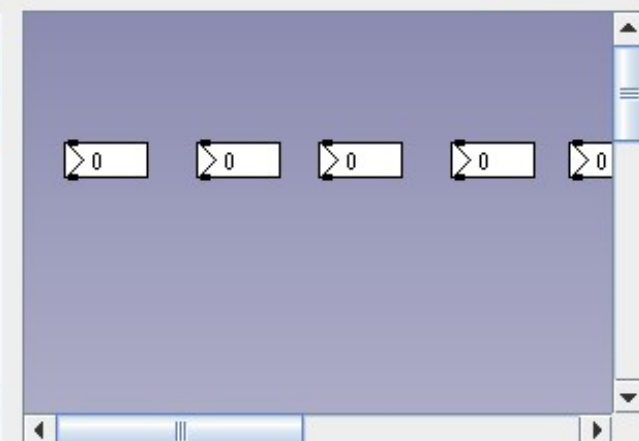
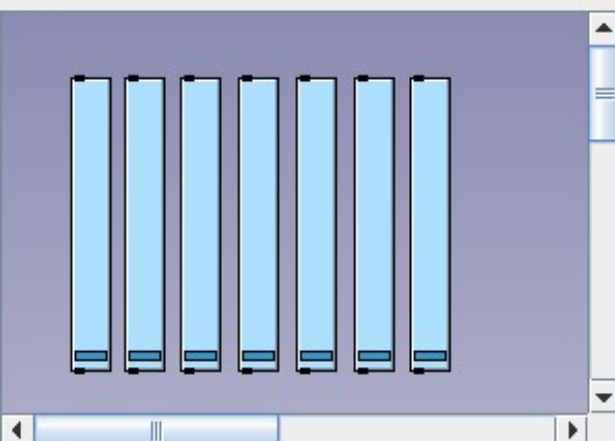
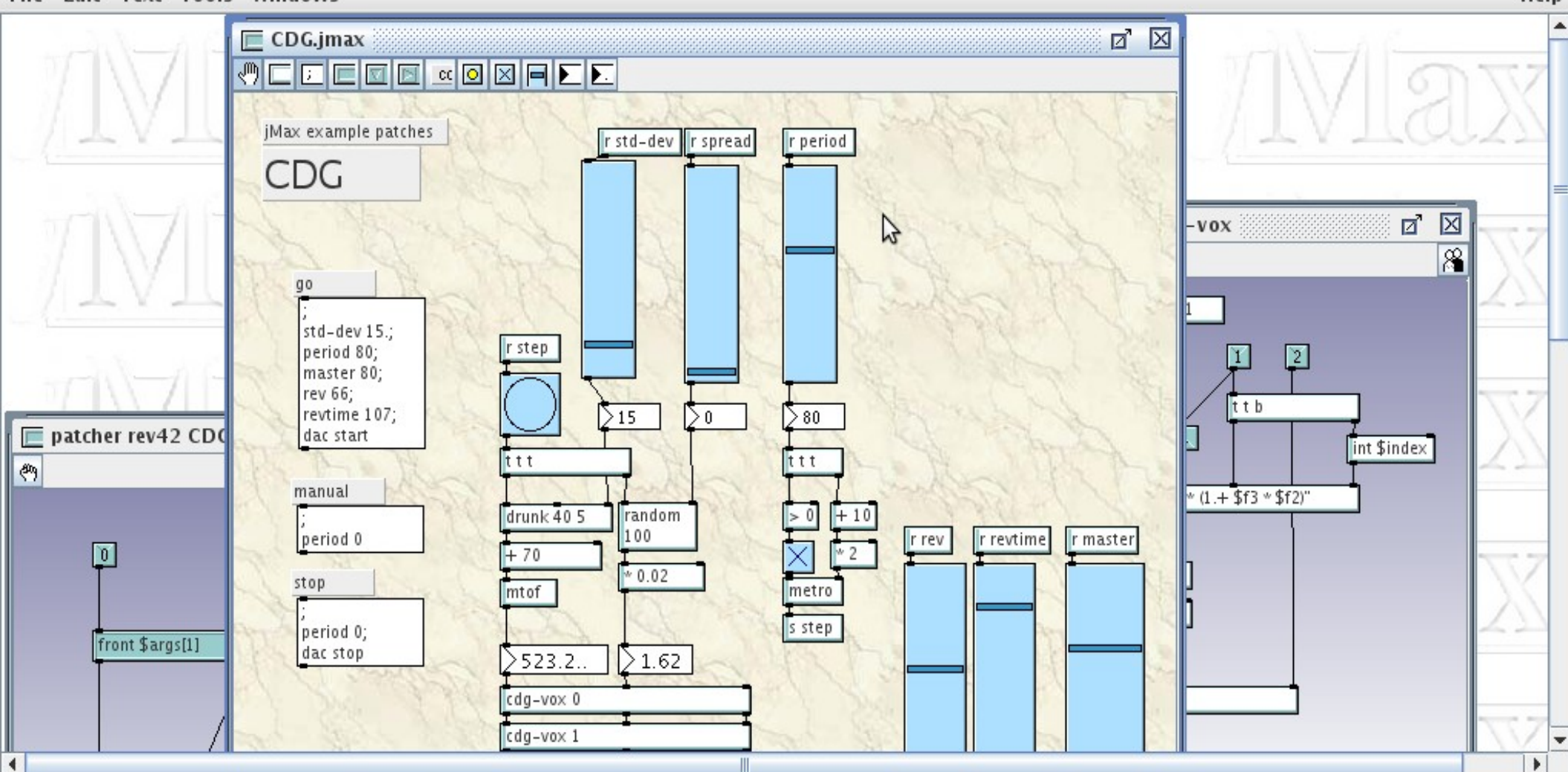
- CDG.jmax
 - cdg-vox 0
 - cdg-vox 1
 - cdg-vox 2
 - cdg-vox 3
 - rev42 CDG
 - feed_in
 - feed_b
 - out
 - front \$

CDG.jmax x patcher rev42 CDG x template cdg-vox x

**jMax** jMax Console x

Copyright (C) 2008 Maurizio De Cecco, Enzo Maggi, Francois Dechelle
Copyright (C) 1994, 1995, 1998, 1999 IRCAM - Centre Georges Pompidou
This is free software with ABSOLUTELY NO WARRANTY.
See file LICENSE for more details.

Show: Editable



UI: Work In Progress (1/2)

- XML configuration
 - A simple, central repository
 - Storing declarative configuration data
 - Loaded from XML files
 - Progressively substituting the tcl based configuration
- Generic object property inspector
 - Reduces object specific java code
 - Requires a change in the plumbing

UI: Work In Progress (2/2)

- Redefinition of the jMax packages
 - Using an explicit XML declarative description
 - Single file, zipped packages
 - Retrieval of packages from a network repository
- Projets
 - Projets will be application specific packages
 - UI to be defined (but simple)

FTS and DSP: Future Plan (1/5)

- The current execution model is limited:
 - Poll input/compute/output loop
 - Single threaded
 - Soft real time
 - Does not scale on multi-core systems
 - Does not fit within callback based audio system (jack)
 - Supports limited parallelism:
 - Disk streaming
 - UI vs. control/DSP computation

FTS and DSP: Future Plan (2/5)

- A multi threaded execution model
 - One thread for each event source
 - Explicit synchronisation on global resources
 - Different priorities for different event sources
 - DSP computation
 - Real time control events (MIDI, timers)
 - UI events
 - Parallel extension of the control language
 - Fork, join, mutex objects

FTS and DSP: Future Plan (3/5)

- Multicore architectures: a strategic objective
 - Entry level: 4 cores, 8 threads
 - Sweet spot: 2 sockets, 8 cores, 16 threads
 - High end: 4 sockets, 16-32 cores, 32-64 threads
- Exploiting this level of parallelism requires
 - Radical changes of the current computational model
 - A partial or total paradigm shift
 - Very good coding

FTS and DSP: Future Plan (4/5)

- The new FTS DSP system
 - Generic with respect to data size and type
 - Generic with respect to tick frequency
- Graph level optimizer
 - Type inference
 - Standard optimisations
 - Dead code elimination
 - Constant propagation
 - Aggregation of operations (ex. multiply-add)

FTS and DSP: Future Plan (5/5)

- Data Flow computational engine
 - Fully parallel operations
 - Synchronisation on data availability
 - Grouping of *similar operations* to reduce overhead
 - Thread pool for execution
 - Automatic load balancing, multi-scalar and pipelined execution
 - Supports conditional execution of sub patches
 - Requires the careful coding of a non blocking data flow kernel

Projet Information

- Home page: <http://www.jmax-phoenix.org>
 - Hosts the main page and a wiki
- Sourceforge site:
 - <http://sourceforge.net/projects/jmax-phoenix/>
 - Developers mailing list
 - Users mailing list
 - Binary snapshots
- License:
 - Currently GPL with exception for loaded packages

Call For Developers !!

- We Need Help !!
 - Core developers (C, Objective C, Java, Swing)
 - Object developers (DSP, Control)
 - Developers to port existing libraries
 - Testers and users
 - A graphic designer !!
- Contacts:
 - contacts@jmax-phoenix.org

Q&A